A Constructive Proof of Open Induction Using Delimited Control Operators

Danko Ilik[1] and Keiko Nakata[2]

[1] danko.ilik@gmail.com
[2] Institute of Cybernetics, Tallinn University of Technology, Tallinn, Estonia
keiko@cs.ioc.ee

Abstract. Open Induction on Cantor space (OI) is a principle classically equivalent to Dependent Choice, but, unlike the later, it is closed under double-negation translation and A-translation. In the context of Constructive Reverse Mathematics, Wim Veldman has shown that, in presence of Markov's Principle (MP), OI is equivalent to Double-negation Shift (DNS). In this paper, we use the proof of Veldman, and our previous work on interpreting DNS by delimited control operators, to give a constructive proof of OI using delimited control operators. We also point out that while the use of MP can be replaced by strengthening the DNS schema, one does have to use a countable axiom of choice in the proof.

Keywords: open induction, axiom of choice, computational interpretation, constructive proof, delimited control operators, realizability

1 Introduction

Let A be an open subset of Cantor space, $\mathbb{N} \to \{0,1\}$, that is, let A be defined by an enumeration (p_n) of finite bit-strings such that $\alpha \in A$ stands for $\exists n,k \in \mathbb{N}(\overline{\alpha}n = p_k)$, where $\overline{\alpha}n$ is the finite initial segment of length n of the bit-stream α. Define the lexicographic order $\beta < \alpha$ on infinite bit-streams by $\exists n \in \mathbb{N}(\overline{\beta}n = \overline{\alpha}n \wedge (\beta(n) = 0 \wedge \alpha(n) = 1))$.

Then, the principle of Open Induction on Cantor space is the following statement:

$$\forall \alpha\, (\forall \beta < \alpha\, (\beta \in A) \to \alpha \in A) \to \forall \alpha (\alpha \in A). \qquad \text{(OI)}$$

In other words, all infinite paths α through the infinite binary tree that is Cantor space can be shown to satisfy an open property A, if A is progressive: a path α is in A when all paths β "left" of α are in A. Note that A can be uncountable and that $<$ is not a well-founded order on infinite paths, in general.

The principle OI was isolated by Raoult [14] who using it give a version of Nash-Williams' proof of Kruskal's theorem that does not use the Axiom of Choice. Later, it was introduced to the field of Constructive Mathematics by Coquand [3], who also gave a justification of it in terms of bar induction [4]. Berger [2] showed that it is classically equivalent to the Axiom of Dependent Choice (DC), and that, unlike DC, it is closed under double-negation- and A-translation; this means, in particular, that there is a simple way to extract intuitionistic proofs (and programs) from classical proofs of Σ^0_1-statements that use OI.

In Constructive Reverse Mathematics, Veldman investigated the relationships between various (more-than-)intuitionistic principles [18,17], and, among others, proved the equivalence of OI and Double-negation Shift (DNS) in presence of Markov's Principle (MP). Given that it is possible to obtain proofs for both MP [8] and DNS [10] using constructive logical systems based on delimited control operators, it was a natural next step to attempt to provide a constructive proof of OI based on delimited control operators.

This paper is organized as follows. In Section 2, we revise Veldman's proof of OI from DNS and MP that uses a new logical principle, EnDec, as a stepping stone. In Section 3, we revise the logical system from [10] and use its ability to prove DNS_S (a slight strengthening of plain DNS) using delimited control operators, in order to prove EnDec without explicitly using MP. In Section 4, we give a formalized proof term for OI in a variant of HA^ω based on the logical system from the previous section. In the final Section 5, we discuss future work and mention related works.

2 From DNS and MP to Open Induction

Notation Unless otherwise noted, the metavariables $l, m, n\ k$ range over natural numbers, $\mathbb{N}$. We use p, q to denote finite bit-strings, B^*, and α, β for infinite bit-streams, $\mathbb{N} \to \mathrm{B}$. We omit type annotations for these metavariables. For $p, q : \mathrm{B}^*$, $p * q$ denotes the concatenation of p and q. For $\alpha : \mathbb{N} \to \mathrm{B}$ and $n : \mathbb{N}$, $\overline{\alpha}n$ denotes the finite initial segment of length n of α. Analogously, for $p : \mathbb{N} \to \mathrm{B}$ and $n : \mathbb{N}$, $\overline{p}n$ denotes the initial segment of length n of p (if n is less or equal to the length of p, otherwise it is undefined). Concrete bit-strings are constructed using the notation $\langle\cdot\rangle$; $p * \langle 0 \rangle$ thus means that a zero bit has been appended at the end of p.

The notation $\beta < \alpha$ abbreviates $\exists n(\overline{\beta}n = \overline{\beta}n \wedge \beta(n) = 0 \wedge \alpha(n) = 1)$. We abbreviate $(A_1 \to A_2) \wedge (A_2 \to A_1)$ to $(A_1 \leftrightarrow A_2)$.

Consider the following principle.

Axiom 1 (EnDec). For any $B \subseteq \mathbb{N}$ which is recursively enumerable, if

> for any $A \subseteq \mathbb{N}$ which is decidable and such that $A \subseteq B$, we have that if $\exists m(m \notin A)$ then $\exists m(m \notin A \wedge m \in B)$,

then $\mathbb{N} \subseteq B$ (and hence B is decidable).

In this section, we derive Open Induction on Cantor space from Double-negation Shift and Markov's Principle, by adapting the argument by Veldman [18] of Open Induction on the closed unit interval $[0,1]$[3]. EnDec is a stepping stone.

Theorem 1. *EnDec implies Open Induction on Cantor space.*

Proof. Let A be an open subset of Cantor space, i.e. let $\pi : \mathbb{N} \to \mathrm{B}^*$ be an enumeration of finite bit-strings so that, for any α, $\alpha \in A$ is a notation for $\exists l, m(\overline{\alpha}l = \pi m)$[4]. Let also A be *progressive*, that is,

$$\forall \alpha(\forall \beta < \alpha(\beta \in A) \to \alpha \in A).$$

We want to show that $\forall \alpha(\alpha \in A)$. It suffices to show $\langle\rangle \in B$ for the empty bit-string $\langle\rangle$, where B is defined as

$$p \in B \text{ iff } \exists k \forall q \in \mathrm{B}^k \exists l, m(\overline{p * q}\, l = \pi\, m).$$

We show that B is equal to B^*, using EnDec. Notice that B^* is countable and B is recursively enumerable[5]. It is left to to show that, for any decidable subset $C \subseteq B$, if $\exists q(q \notin C)$, then $\exists r(r \notin C \wedge r \in B)$.

Suppose that such C and q are given. If $\langle\rangle \in C \subseteq B$, then we have that $q \in B$. So we are done. We assume $\langle\rangle \notin C$. Since C is decidable, we can define α such that

$$\alpha(n) := \begin{cases} 0 \text{ , if } \overline{\alpha}n * \langle 0\rangle \notin C \\ 1 \text{ , if } \overline{\alpha}n * \langle 0\rangle \in C \text{ and } \overline{\alpha}n * \langle 1\rangle \notin C \\ 0 \text{ , if } \overline{\alpha}n * \langle 0\rangle \in C \text{ and } \overline{\alpha}n * \langle 1\rangle \in C \end{cases}$$

[3] Veldman showed the *equivalence* of OI and EnDec, but in this paper we are just interested in one direction of the proof.

[4] The progressiveness ensures that A is non-empty. Hence we may take π to be a total function.

[5] B is recursively enumerable because it is defined by a Σ^0_1-formula: the bounded universal quantifier does not pose a problem, since it could be interpreted as a bounded minimization operator, for example like in §3.5 of [11].

The bit-stream α tries to stay outside of C for as long as possible, first trying to "turn left" (0) before "turning right" (1). If that is not possible, not only is $\overline{\alpha}(n+1) \in C \subseteq B$, but, thanks to the definition of B, also $\overline{\alpha}n \in B$.

Now, we see that it is not difficult to find an initial segment of α that is both not in C and in B: we follow α up to the first point where it enters B – but that is if ever α enters into B. To have a guarantee of that, we show that $\alpha \in A$ using progressivity of A. Let $\beta < \alpha$ i.e. $\exists n(\overline{\beta}n = \overline{\alpha}n \wedge \beta(n) = 0 \wedge \alpha(n) = 1)$. By construction of α, $\alpha(n) = 1$ can only be the case if both $\overline{\beta}(n+1) = (\overline{\beta}n) * \langle 0 \rangle = (\overline{\alpha}n) * \langle 0 \rangle \in C$ and $\overline{\alpha}(n+1) = (\overline{\alpha}n) * \langle 1 \rangle \notin C$. Because $\overline{\beta}(n+1) \in C \subseteq B$, we have that $\beta \in A$, which was to be shown.

From $\alpha \in A$ we obtain l, m such that $\overline{\alpha}l = \pi m$, and we can finish the proof by induction, proving

$$\forall n \leq l \left(\overline{\alpha}(l-n) \notin C \to \exists l'(\overline{\alpha}l' \notin C \wedge \overline{\alpha}l' \in B)\right).$$

In the base case, $n = 0$, we have by hypothesis that $\overline{\alpha}l \notin C$, and we have from $\alpha \in A$ that $\overline{\alpha}l \in B$; so we set $l' := l$. In the induction case for $n+1$ we consider three possibilities:

1. if $\overline{\alpha}(l-(n+1)) * \langle 0 \rangle \notin C$, then $\overline{\alpha}(l-(n+1)) * 0 = \overline{\alpha}(l-(n+1)+1) = \overline{\alpha}(l-n) \notin C$ and we can use the induction hypothesis;
2. if $\overline{\alpha}(l-(n+1)) * \langle 0 \rangle \in C$ and $\overline{\alpha}(l-(n+1)) * \langle 1 \rangle \notin C$, then $\overline{\alpha}(l-(n+1)) * \langle 1 \rangle = \overline{\alpha}(l-(n+1)+1) = \overline{\alpha}(l-n)$ and we can use the induction hypothesis;
3. if $\overline{\alpha}(l-(n+1)) * \langle 0 \rangle \in C$ and $\overline{\alpha}(l-(n+1)) * \langle 1 \rangle \in C$, as discussed above, we get that $\overline{\alpha}(l-(n+1)) \in B$ and we can set $l' := l-(n+1)$, because by hypothesis we also have that $\overline{\alpha}(l-(n+1)) \notin C$.

The first two cases could be merged into one, verifying only whether $\overline{\alpha}(l-(n+1)+1) \notin C$. □

Remark 1. In the previous proof, we implicitly used an axiom of countable choice, $\mathrm{AC}^{0,0}$, in constructing the sequence α from the decidability of C. This use becomes apparent in the formalization of Section 4. The fact that we use $\mathrm{AC}^{0,0}$ is important: since MP and DNS are classical theorems, not using $\mathrm{AC}^{0,0}$ would mean that we have reduced OI (and DC) to plain classical logic without choice.

The following principle of Double-negation Shift (DNS) has been isolated by Spector [15] as sufficient for extending Gödel's functional

interpretation from Arithmetic to Analysis, thanks to the fact that DNS proves the double-negation translation of the axiom of countable choice.

Axiom 2 (DNS). $\forall n \neg\neg A(n) \to \neg\neg \forall n A(n)$, for any formula $A(n)$.

Veldman uses the following version of DNS.

Axiom 3 (DNS$^{\mathrm{V}}$). $\neg\neg\forall n(A(n) \vee \neg A(n))$, for any formula $A(n)$.

Remark 2. The proof of equivalence of 2 and 3 is analogous to the proof of equivalence between the law of double-negation elimination (DNE) and the law of excluded middle (EM). Note, however, that in minimal logic, which is intuitionistic logic without the rule of $\bot$-elimination (*ex falso quodlibet*), EM is weaker than DNE [1]. We expect a similar result for DNS, i.e., that DNS$^{\mathrm{V}}$is weaker than DNS in minimal logic.

Markov's Principle is the following axiom schema.

Axiom 4 (MP). For any Σ^0_1-formula S, we have that $\neg\neg S \to S$.

Theorem 2. *DNS^{V}and MP together imply EnDec.*

Proof. Let the premises of EnDec hold. Given $n \in \mathbb{N}$, we have to prove $n \in B$, which is a Σ^0_1-formula. We are entitled to apply MP. Now, we have to show that $\neg\neg(n \in B)$. Suppose $\neg(n \in B)$. Thanks to DNS$^{\mathrm{V}}$, it suffices to prove $\bot$ assuming moreover that B is decidable, i.e., $\forall n(n \in B \vee \neg(n \in B))$. We make a case distinction. If $n \in B$ we use the hypothesis $\neg(n \in B)$ to derive $\bot$. If $\neg(n \in B)$, we use the premise of EnDec by taking $A := B$ (remember B is now assumed decidable). This gives us $\exists m(m \in B \wedge \neg(m \in B))$, from which we derive $\bot$. □

3 A Constructive Logic Proving EnDec

In this section, we revise the logical system MQC$_+$of [10] and show that one can prove in it EnDec without an explicit use of MP, thanks to the slightly stronger form of DNS that it proves.

MQC$_+$ is a pure predicate logic system, that, in addition to the usual rules of minimal intuitionistic predicate logic, adds two rules for proving Σ-formulas (formulas without $\forall$ and $\to$). The rule "reset",

$$\frac{\Gamma \vdash_S S}{\Gamma \vdash_\diamond S}\ \#\ \text{(“reset”)},$$

allows one to set a marker (under the turnstile) meaning that one wants to prove a Σ-formula S. Once the marker is set, one can use the “shift” rule,

$$\frac{\Gamma, A \Rightarrow S \vdash_S S}{\Gamma \vdash_S A}\ \mathscr{S}\ \text{(“shift”)},$$

to prove by a principle related to double-negation elimination from classical logic. The idea is to internalize in the formal system the fact, known from Friedman-Dragalin's A-translation, that a classical proof of a Σ^0_1-formula can be translated to an intuitionistic proof of the same formula, showing that classical proofs of such formulas are in fact constructive. The first system built around this internalization idea was Herbelin's [8] with the power to derive Markov's Principle. It satisfies, like MQC_+, the disjunction and existence properties characteristic of plain intuitionistic logic.

The names “shift” and “reset” come from the computational intention behind the normalization of these proof rules, Danvy-Filinski's delimited control operators [5,6,7]. These operators were developed in the theory of programming languages with the aim of enabling to write in so-called *direct style* every continuation-passing style (CPS) program. Since CPS transformations are known to be one and the same thing as double-negation translations [13], one can think of shift/reset in Logic as enabling to prove *directly* theorems whose double-negation translation is intuitionistically provable. In order for this facility to remain constructive, we allow its use only for proving Σ-formulas.

The natural deduction system, with proof term annotations, is given in Table 3. The diamond in the subscript of $\vdash$ is a wild-card. The usual intuitionistic rules neither “read” nor “write” this marker. The reset rule is the one that sets it – if it is not already set, in which case the formula S must be the same below and above the line (this kind of use of reset would have no logical purpose, but it would affect the course of normalization). The rule shift is the one that must be assured that we are ultimately proving a Σ-formula, and that is why it can only be applied when the marker is set.

The normalization proof for MQC_+ relies on the fact that at most one Σ-formula is used globally, so, although we can have multiple uses

$$\frac{(a:A)\in\Gamma}{\Gamma\vdash_\diamond a:A}\ \mathrm{Ax}$$

$$\frac{\Gamma\vdash_\diamond p:A_1 \qquad \Gamma\vdash_\diamond q:A_2}{\Gamma\vdash_\diamond (p,q):A_1\wedge A_2}\ \wedge_I \qquad\qquad \frac{\Gamma\vdash_\diamond p:A_1\wedge A_2}{\Gamma\vdash_\diamond \pi_i p:A_i}\ \wedge_E^i$$

$$\frac{\Gamma\vdash_\diamond p:A_i}{\Gamma\vdash_\diamond \iota_i p:A_1\vee A_2}\ \vee_I^i$$

$$\frac{\Gamma\vdash_\diamond p:A_1\vee A_2 \qquad \Gamma,a_1:A_1\vdash_\diamond q_1:C \qquad \Gamma,a_2:A_2\vdash_\diamond q_2:C}{\Gamma\vdash_\diamond \mathsf{case}\ p\ \mathsf{of}\ (a_1.q_1\|a_2.q_2):C}\ \vee_E$$

$$\frac{\Gamma,a:A_1\vdash_\diamond p:A_2}{\Gamma\vdash_\diamond \lambda a.p:A_1\to A_2}\ \to_I \qquad\qquad \frac{\Gamma\vdash_\diamond p:A_1\to A_2 \qquad \Gamma\vdash_\diamond q:A_1}{\Gamma\vdash_\diamond pq:A_2}\ \to_E$$

$$\frac{\Gamma\vdash_\diamond p:A(x) \qquad x\text{-fresh}}{\Gamma\vdash_\diamond \tilde{\lambda}x.p:\forall xA(x)}\ \forall_I \qquad\qquad \frac{\Gamma\vdash_\diamond p:\forall xA(x)}{\Gamma\vdash_\diamond pt:A(t)}\ \forall_E$$

$$\frac{\Gamma\vdash_\diamond p:A(t)}{\Gamma\vdash_\diamond (t,p):\exists x.A(x)}\ \exists_I$$

$$\frac{\Gamma\vdash_\diamond p:\exists x.A(x) \qquad \Gamma,a:A(x)\vdash_\diamond q:C \qquad x\text{-fresh}}{\Gamma\vdash_\diamond \mathsf{dest}\ p\ \mathsf{as}\ (x.a)\ \mathsf{in}\ q:C}\ \exists_E$$

$$\frac{\Gamma\vdash_S p:S}{\Gamma\vdash_\diamond \#p:S}\ \#\ (\text{“reset”}) \qquad\qquad \frac{\Gamma,k:A\to S\vdash_S p:S}{\Gamma\vdash_S \mathscr{S}k.p:A}\ \mathscr{S}\ (\text{“shift”})$$

where S denotes a Σ-formula

Table 1. Natural deduction system for MQC_+ with proof terms annotating the rules

of shift and reset in a derivation, all resets must agree on the formula to be set at. This suffices for the purpose of the present paper, although in future we hope to extend to normalization proof so that we remove this limitation.

The following fact shows the utility of proving with shift and reset.

Theorem 3. *Let S be a Σ-formula (a $\forall$-$\to$-free formula) and $A(x)$ an arbitrary formula. The following version of DNS$^{\mathrm{V}}$,*

$$((\forall x(A(x) \vee (A(x) \to S))) \to S) \to S, \qquad (\mathrm{DNS}^{\mathrm{V}}_S)$$

is provable in MQC_+.

Proof. Using the proof term $\lambda h.\#h(\tilde{\lambda}x.\mathscr{S}k.k(\iota_2(\lambda a.k(\iota_1 a))))$. □

This version of $\mathrm{DNS}^{\mathrm{V}}$ already has some form of MP built in, as can be seen from the proof of Theorem 4.

We now state a version of EnDec which is suitable for use in minimal logic, where $\bot$-elimination is absent.

Axiom 5 (A minimal-logic version of Axiom 1). For any $B \subseteq \mathbb{N}$ which is recursively enumerable, if

> for any $s \in \mathbb{N}$ and any $A \subseteq \mathbb{N}$ for which $\forall x(x \in A \vee (x \in A \to s \in B))$, and such that $A \subseteq B$, we have that if $\exists m(m \in A \to s \in B)$ then $\exists m((m \in A \to s \in B) \wedge m \in B)$,

then $\mathbb{N} \subseteq B$.

The following result is the minimal-logic analogue of Theorem 2.

Theorem 4. *In MQC_+, one can derive Axiom 5.*

Proof. Let the premises of Axiom 5 hold and let $n \in \mathbb{N}$. To show that $n \in B$, which is a Σ^0_1-formula, we use $\mathrm{DNS}^{\mathrm{V}}_S$ for $A(x) := x \in B$ and $S := n \in B$. Now, given $\forall x(x \in B \vee (x \in B \to n \in B))$, we have to show $n \in B$. We use the premise of Axiom 5 for $s := n$ and $A := B$, and, using the trivial proof of $\exists m(m \in B \to n \in B)$ for $m := n$, the premise gives us a proof of $\exists m(m \in B \wedge (m \in B \to n \in B))$, from which we derive $n \in B$. □

Now, since the proof of Theorem 1 is purely intuitionistic, it follows that one can combine it with the proof of Theorem 4 to derive OI in MQC_+, using also $\mathrm{AC}^{0,0}$ as noted in Remark 1. A precise statement of this fact is the subject of the next section.

4 A Proof Term for Open Induction

In the previous section, we used arithmetic informally. In this section, we give the proof term for OI that we extracted from the proofs of theorems 1 and 4, formalized in the system HA^{ω}_{+} which is the system of axioms HA^{ω} (from §§1.6.15 of [16]) added on top of the predicate logic MQC_{+}.

First, we take a multi-sorted version of MQC_{+}, that is, given different sorts (denoted σ, ρ, τ), the language is extended with individual variables (denoted x, y, z) of any sort, and, for each sort, with quantifiers and quantifier natural deduction rules for it. We will not annotate the quantifiers with their sorts, since those will be clear from the context; we may annotate the variables by their sort when we want to avoid ambiguity.

The sorts can be built inductively, according to the following rules: there is a sort named 0; if ρ and σ are sorts, then there is a sort named $\rho \to \sigma$. The intended interpretation is that the sort 0 stands for $\mathbb{N}$, the sort $0 \to 0$ stands for functions $\mathbb{N} \to \mathbb{N}$, the sort $((0 \to 0) \to 0)$ for functionals $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$, etc. We will employ the word 'type' instead of sort, henceforth, and we abbreviate the type $0 \to 0$ by 1.

Now, we add to the language a binary predicate symbol $=_0$ for individual terms of type 0, intended to be interpreted as (the decidable) equality on $\mathbb{N}$. The individual terms will be built from the function symbols 0^0 (zero), $(\cdot+1)^1$ (successor), $\Pi^{0\to\tau\to 0}$ and $\Sigma^{(\sigma\to\rho\to 0)\to(\sigma\to\rho)\to\sigma\to 0}$ (combinators), and $\mathrm{R}_0^{0\to 0\to(0\to 0\to 0)\to 0}$ (recursor of type 0). There is also the function symbol of juxtaposition which is not explicitly denoted: for terms $t^{\sigma\to\tau}$ and s^{σ}, ts is a term of type τ.

The axioms defining these symbols are (the universal closures of each of):

$$x =_0 x, \qquad x =_0 y \to y =_0 z \to x =_0 z, \qquad x =_0 y \to x+1 =_0 y+1,$$

$$x =_0 y \to t[x/z] =_0 t[y/z] \qquad \text{where } t[x/z] \text{ is the simultaneous substitution of } x \text{ for } z \text{ in } t$$

$$\begin{aligned} \Pi xy &=_0 x \\ \Sigma xyz &=_0 xz(yz) \\ \mathrm{R}_0 0yz &=_0 y \\ \mathrm{R}_0(x+1)yz &=_0 z(\mathrm{R}_0 xyz)x \end{aligned}$$

We also add the axiom schema of induction, for arbitrary formula $A(x)$:

$$A(0) \to \forall x(A(x) \to A(x+1)) \to \forall x(A(x)) \qquad \text{(IA)}$$

We shall also need the following axiom of countable choice, which, although not part of HA^ω, is an admissible rule for HA^ω:

$$\forall x^0 \exists y^0 A(x,y) \to \exists \phi^1 \forall x^0 A(x, \phi x) \qquad (\mathrm{AC}^{0,0})$$

Since "$=_0$" is the only predicate symbol, all atomic (prime) formulas are of form $t =_0 s$. This allows us to show that $x =_0 y \to A(x) \to A(y)$, by induction on the complexity of formula A.

It is known that using the combinators one may define an individual term for lambda abstraction, denoted $\dot{\lambda} x.t$, of type 1, which satisfies the usual β-reduction axiom, $(\dot{\lambda} x^0.s^0)t^0 =_0 s[t/x]$. Using this and the recursor R_0, one can easily define all the usual primitive recursive functions. Using the thus defined predecessor function, and the induction axiom, one can derive the remaining Peano axioms, $x+1 =_0 y+1 \to x =_0 y$, and $(x+1=0) \to 1=0$, where we took $1=0$ instead $\bot$ because we are in minimal logic. In fact, in the presence of arithmetic, one can prove, again by induction, that the rule of $\bot$-elimination (with $\bot$ replaced by $1=0$) is derivable, although we will not need it.

Some notational conventions follow. We shall need to speak of bits, finite sequences of bits (bit-strings), and infinite sequences of bits (bitstreams). Bits and bitstreams can be encoded by natural numbers, but, instead of using the type 0 for terms of that kind, to be more precise, we will write B and B^*. Bitstreams can be encoded by terms of type $0 \to 0$, but instead of that we will write $0 \to \mathrm{B}$. Individual variables of type B^* will be denoted p, q, r, s (possibly with sub/super-scripts), while those of type $0 \to \mathrm{B}$ will be denoted α, β, χ (possibly with sub/super-scripts). Concrete bit-string terms may be constructed using the notation $\langle \cdot \rangle$, like for example $\langle\rangle$, the empty bit-string, $\langle 0 \rangle$, the bit-string of lenght 1 that contains a 0, $\langle 1,1,1,1 \rangle$, the bit-string that contains four 1-s, etc. We will denote the operation for concatenation of bit-strings p and q by $p * q$. The operations $\overline{p}n$ and $\overline{\alpha}n$ produce the prefixing bit-string of length n of p and α. Although p is not a function, we will use the notation $p(n)$ to extract the $n-1$-st bit of p. The operator $\mathrm{head}(p)$ returns the first bit of p, while $\mathrm{tail}(p)$ returns the string that follows the first bit of p. We will also use the fact that one can define by primitive recursion a

term if $\cdots$ then $\cdots$ else $\cdots$ of type $0 \to 0 \to 0 \to 0$, or, more precisely, $\mathrm{B} \to \mathrm{B} \to \mathrm{B} \to \mathrm{B}$, such that the following equations hold:

$$\begin{aligned} \mathsf{if}\ 0\ \mathsf{then}\ y\ \mathsf{else}\ z &=_0 z \\ \mathsf{if}\ x+1\ \mathsf{then}\ y\ \mathsf{else}\ z &=_0 y \end{aligned}$$

We will also need the usual operation $\max : 0 \to 0 \to 0$ on numbers. All these operations can be defined by a restricted amount of primitive recursion at higher types, level 3 of the Grzegorcyk hierarchy would suffice. Hence we could work in a corresponding subsystem of HA^ω, like for example $\mathrm{G_3A}_i^\omega$ from §3.5 of [11].

Let us now formalize the concepts involved in the proof of OI. An open set U in Cantor space will be given by a term π of type $0 \to 0$, or, more precisely, $0 \to \mathrm{B}^*$, that is, an enumeration of basic opens. Each bit-string πn is a basic open, and membership in the open set, $\alpha \in U$, means that α is covered by some basic open. Formally, we define

$$\alpha \in U \text{ iff } \exists l^0 \exists m^0 (\overline{\alpha} l =_0 \pi m),$$

and we see that membership in U is a Σ-formula. The relation $<$ on bit-streams was already formalized as

$$\beta < \alpha \text{ iff } \exists n^0 \left(\overline{\beta} n =_0 \overline{\alpha} n \wedge (\beta(n) =_0 0 \wedge \alpha(n) =_0 1) \right).$$

To formalize the statement of Axiom 5, we represent the recursively enumerable set B by a Σ-formula $B(x)$, and we represent the decidable subset A by a characteristic function $\chi_A^{\mathrm{B}^* \to \mathrm{B}}$ such that $\chi_A(p) = 1$ iff $p \in A$. We thus obtain the following formula for Axiom 5:

$$\begin{aligned} \forall s^{\mathrm{B}^*} \Big(\forall \chi_A^{0 \to \mathrm{B}} \big(\forall x (\chi_A(x) = 1 \to B(x)) \to \\ \exists q^{\mathrm{B}^*} (\chi_A(q) = 1 \to B(s)) \\ \exists r^{\mathrm{B}^*} ((\chi_A(r) = 1 \to B(s))) \wedge B(r)) \big) \Big) \\ \to \forall p^{\mathrm{B}^*} B(p) \end{aligned}$$

When we use Axiom 1 to prove OI, we do so for the formula

$$B(x) := \exists k^0 \forall q^{\mathrm{B}^k} \exists l^0 \exists m^0 (\overline{x * q} l = \pi m),$$

where $\forall q^{\mathrm{B}^k}$ denotes a *bounded* universal quantifications over bit-strings of length k. Hence, $B(x)$ is still a Σ^0_1-formula, and such that, for any α, $\exists n(\overline{\alpha}n \in B)$ iff $\alpha \in A$.

The formalized proof term for OI is shown in Figure 1. To ease the presentation, at certain places, we have put after a semicolon the type annotation for individual terms, and the formula for proof terms. Some parts, being too long, have been put below the main proof term.

To not obfuscate the proof term with equality-rewriting terms, we suppress the use of equality axioms. It is known that equality proofs have no computational content when extracting programs, as they are realized by singleton data types.

$$
\begin{array}{l}
\tilde{\lambda}\pi : 0 \to \mathrm{B}^* .\lambda h : \forall\alpha(\forall\beta < \alpha(\beta \in A) \to \alpha \in A).\\
(\#\mathsf{dest}\ a_C(\tilde{\lambda}x.\mathscr{S}k.k(\iota_2(\lambda a.k(\iota_1 a))))\ \mathsf{as}\ (\chi.b)\ \mathsf{in}\\
\quad \mathsf{dest}\ (h\alpha(\tilde{\lambda}\beta.\lambda(h' : \beta < \alpha).\\
\quad\quad \mathsf{dest}\ (h' : \beta < \alpha)\ \mathsf{as}\ (n.h'')\ \mathsf{in}\\
\quad\quad \mathsf{dest}\ (a_1(\pi_2(\pi_2 h'')) : \overline{\beta}(n+1) \in B)\ \mathsf{as}\ (k.h''')\ \mathsf{in}\\
\quad\quad h'''(\langle\beta(n+1)\rangle * \cdots * \langle\beta(n+k)\rangle)))) : \alpha \in A)\ \mathsf{as}\ (l.c)\ \mathsf{in}\\
\quad \mathsf{dest}\ c\ \mathsf{as}\ (m.d)\ \mathsf{in}\\
\quad\quad a_I\,(\lambda h.h)\,a_3\,l\,(0,\lambda^{<}q.(l,(m,d))))
\end{array}
$$

$$
\begin{array}{l}
\alpha := \dot{\lambda}n.\\
\quad \mathrm{R}_0(n+1,\langle\rangle,(\dot{\lambda}z.\dot{\lambda}n'.z * \langle\mathsf{if}\ \chi(z*\langle 0\rangle)\ \mathsf{then}\ (\mathsf{if}\ \chi(z*\langle 1\rangle)\ \mathsf{then}\ 0\ \mathsf{else}\ 1)\ \mathsf{else}\ 0\rangle))(n)\\
a_1 : \alpha(n) = 1 \to \overline{\beta}(n+1) \in B := \lambda h.\mathsf{case}\ a_B(\chi(\overline{\beta}(n+1)))\ \mathsf{of}\\
\quad \Big(h_1.\mathscr{S}k.(\pi_1(\pi_2(b(\overline{\beta(n+1)})))\,h_1)\,(\pi_1(\pi_1(b(\overline{\beta(n+1)})))\,h_1)\|h_2.\pi_1(\pi_1(b(\overline{\beta(n+1)})))\,h_2\Big)\\
a_3 := \lambda n.\lambda h_I : \overline{\alpha}n \in B \to \langle\rangle \in B.\lambda h : \overline{\alpha}(n+1) \in B.\\
\quad \mathsf{case}\ a_B(\chi(\overline{\alpha}n * \langle 0\rangle))\ \mathsf{of}\\
\quad\quad (h_1.(\pi_1\pi_2(b(\overline{\alpha}(n+1)))\,h_1)(\pi_1\pi_1(b(\overline{\alpha}(n+1)))\,h)\\
\quad\quad \|h_2.\mathsf{case}\ (a_B(\chi(\overline{\alpha}n * \langle 1\rangle)))\ \mathsf{of}\\
\quad\quad\quad (h_{21}.(\pi_1\pi_2(b(\overline{\alpha}(n+1)))\,h_{21})(\pi_1\pi_1(b(\overline{\alpha}(n+1)))\,h)\\
\quad\quad\quad \|h_{22}.h_I\,a_4))\\
a_4 : \overline{\alpha}n \in B :=\\
\mathsf{dest}\ ((\pi_1\pi_1(b(\overline{\alpha}n * \langle 0\rangle)))\,h_2 : \overline{\alpha}n * \langle 0\rangle \in B)\ \mathsf{as}\ (k_0.f_0 : \forall q : \mathrm{B}^{k_0}.\exists l,m(\overline{\overline{\alpha}n * \langle 0\rangle * q}\ l = \pi m))\ \mathsf{in}\\
\mathsf{dest}\ ((\pi_1\pi_1(b(\overline{\alpha}n * \langle 1\rangle)))\,h_{22} : \overline{\alpha}n * \langle 1\rangle \in B)\ \mathsf{as}\ (k_1.f_1 : \forall q : \mathrm{B}^{k_1}.\exists l,m(\overline{\overline{\alpha}n * \langle 1\rangle * q}\ l = \pi m))\ \mathsf{in}\\
(\max(k_0,k_1)+1, \lambda q : \mathrm{B}^{\max(k_0,k_1)+1}.\mathsf{if}\ \mathrm{head}(q)\ \mathsf{then}\ f_1(\overline{\mathrm{tail}(q)}k_1)\ \mathsf{else}\ f_0(\overline{\mathrm{tail}(q)}k_0)
\end{array}
$$

Fig. 1. Proof term for OI

We explain the behaviour of the proof term. Given an enumeration π to represent the open set A, and a proof h that A is progressive, it has to show that $\langle\rangle \in B$ – from this, $\forall\alpha(\alpha \in A)$ immediately follows.

To show $\langle\rangle \in B$, which is a Σ-formula, it applies a reset #, and now it has to show the same formula, but classical logic in the form of the shift rule can be used. Indeed, the proof term $\tilde{\lambda}x.\mathscr{S}k.k(\iota_2(\lambda a.k(\iota_1 a)))$ proves the "decidability" of B: $\forall x(x \in B \vee (x \in B \to \langle\rangle \in B))$. Using the proof term a_C for the formula

$$\forall x(x \in B \vee (x \in B \to \langle\rangle \in B)) \to \\ \exists \chi \forall x((\chi(x) = 1 \leftrightarrow x \in B) \wedge (\chi(x) = 0 \leftrightarrow (x \in B \to \langle\rangle \in B))),$$

we obtain from the decidability, a characteristic function $\chi^{\mathrm{B}^* \to \mathrm{B}}$ for B. The proof term a_C is obtained by combining the proof term behind $\mathrm{AC}^{0,0}$ together with a proof term that eliminates disjunction in presence of arithmetic[6]. The proof term b proves the characteristic property of χ.

Now, using this χ, the bit-stream α that we saw in the proof of Theorem 1 can be constructed using R_0 and if $\cdots$ then $\cdots$ else $\cdots$. Next, one needs to show that $\alpha \in A$. Because, that means that one has found the length l at which $\overline{\alpha}l$ is covered by the basic open πm, and then one can show that $\overline{\alpha}0 = \langle\rangle$ is in B. This last fact is derived by the proof term

$$a_I\,(\lambda h.h)\,a_3\,l\,(0, \lambda^{<} q.(l,(m,d))),$$

where a_I is a proof term behind an instance of the induction axiom showing $\forall l(\overline{\alpha}l \in B \to \langle\rangle \in B)$. The proof term a_I uses the proof term a_3 which derives

$$\forall n((\overline{\alpha}\,n \in B \to \langle\rangle \in B) \to \overline{\alpha}\,(n+1) \in B \to \langle\rangle \in B)$$

a_3 is proved by case analysis, considering all four possibilities for the pair $(\chi(\overline{\alpha}n * \langle 0\rangle), \chi(\overline{\alpha}n * \langle 1\rangle))$; in one of the cases, the induction hypothesis h_I needs to be used together with the proof term a_4 that shows that $\overline{\alpha}n \in B$ using the fact that $\overline{\alpha}(n+1) \in B$. To generate the disjunction needed for the case analysis, one uses a proof term a_B for $\forall x^{\mathrm{B}}(x = 0 \vee x = 1)$.

It rests to show that $\alpha \in A$. One uses progressivity h: from β and a proof h' of $\beta < \alpha$, one extracts n and a proof h'' of

$$\overline{\beta}n =_0 \overline{\alpha}n \wedge (\beta(n) =_0 0 \wedge \alpha(n) =_0 1).$$

[6] For the proof of this statement, $(A \vee B) \leftrightarrow \exists x((x = 1 \leftrightarrow A) \wedge (x = 0 \leftrightarrow B))$, see for example §§1.3.7 of [16].

Then, $\pi_2\pi_2 h''$ shows $\alpha(n) =_0 1$, and it is for a_1 to show that $\overline{\alpha}n * \langle 0\rangle = \overline{\beta}(n+1)$ is in B, which shows, with the help of h''', that $\beta \in A$, which concludes the proof[7].

The proof term a_1 derives $\overline{\beta}(n+1) \in B$ from $\alpha(n) = 1$ by making a case distinction. For the first case in which $\chi(\overline{\beta}(n+1)) = 0$, we have an absurdity $0 = 1$. We close the case by the shift rule where the proof term h_1 is used both as of type $\chi(\overline{\beta}(n+1)) = 0$ and $\chi(\overline{\beta}(n+1)) = 1$[8].

5 Conclusion

The analysis of the computational behaviour of the proof term on concrete examples is future work. To do that completely formally, one needs to extract a realizing proof term of $\mathrm{AC}^{0,0}$. While that is a simple matter for HA^{ω}, where one can use Kreisel's modified realizability, we would need a different realizability interpretation for HA^{ω}_{+}, since MP is false in modified realizability.

Also, although *a priori* our proof term is different from Berger's realizer for OI [2], in the future, we would like to verify whether they behave differently operationally. We remark that Berger works with a slightly more general form of open set than Coquand and we do.

We would also like to investigate whether we could adapt a proof term for $\mathrm{AC}^{0,0}$ similar to Herbelin's [9]. His interpretation is compatible with classical logic and is given in a version of Martin-Löf type theory with a weakened version of dependent sums.

Related is also the research program on realizability in classical logic of Krivine [12], who uses (non-delimited) control operators to built new models of classical set theory.

Acknowledgements

We would like to thank Wim Veldman for explaining us some of his results.

[7] The proof term $a_1(\pi_2(\pi_2 h''))$ proves $\overline{\alpha}n * \langle 0\rangle \in B$, from which $\overline{\beta}(n+1) \in B$ follows using equality axioms. As remarked earlier, equality-rewriting is implicit in the proof term.

[8] Alternatively, we could have deduced $\overline{\beta}(n+1) \in B$ from $0 = 1$ by $\bot$-elimination.

References

1. Zena M. Ariola and Hugo Herbelin. Minimal classical logic and control operators. In *Thirtieth International Colloquium on Automata, Languages and Programming, ICALP '03, Eindhoven, The Netherlands, June 30 - July 4, 2003*, volume 2719 of *Lecture Notes in Computer Science*, pages 871–885. Springer, 2003.
2. Ulrich Berger. A computational interpretation of open induction. In F. Titsworth, editor, *Proceedings of the Ninetenth Annual IEEE Symposium on Logic in Computer Science*, pages 326–334. IEEE Computer Society, 2004.
3. Thierry Coquand. Constructive topology and combinatorics. In J. Myers and Michael O'Donnell, editors, *Constructivity in Computer Science*, volume 613 of *Lecture Notes in Computer Science*, pages 159–164. Springer Berlin / Heidelberg, 1992. 10.1007/BFb0021089.
4. Thierry Coquand. A note on the open induction principle, 1997.
5. Olivier Danvy and Andrzej Filinski. A functional abstraction of typed contexts. Technical report, Computer Science Department, University of Copenhagen, 1989. DIKU Rapport 89/12.
6. Olivier Danvy and Andrzej Filinski. Abstracting control. In *LISP and Functional Programming*, pages 151–160, 1990.
7. Olivier Danvy and Andrzej Filinski. Representing control: A study of the CPS transformation. *Mathematical Structures in Computer Science*, 2(4):361–391, 1992.
8. Hugo Herbelin. An intuitionistic logic that proves Markov's principle. In *Proceedings, 25th Annual IEEE Symposium on Logic in Computer Science (LICS '10), Edinburgh, UK, 11-14 July 2010*, page N/A. IEEE Computer Society Press, 2010.
9. Hugo Herbelin. A constructive proof of dependent choice, compatible with classical logic. In *Proceedings of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2012, 25-28 June 2012, Dubrovnik, Croatia*, pages 365–374. IEEE Computer Society, 2012.
10. Danko Ilik. Delimited control operators prove double-negation shift. *Annals of Pure and Applied Logic*, 163(11):1549 – 1559, 2012.
11. U. Kohlenbach. *Applied proof theory: proof interpretations and their use in mathematics*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2008.
12. Jean-Louis Krivine. Dependent choice, 'quote' and the clock. *Theor. Comput. Sci.*, 308(1-3):259–276, 2003.
13. Chetan Murthy. *Extracting Classical Content from Classical Proofs*. PhD thesis, Department of Computer Science, Cornell University, 1990.
14. Jean-Claude Raoult. Proving open properties by induction. *Information Processing Letters*, 29:19–23, 1988.
15. Clifford Spector. Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles formulated in current intuitionistic mathematics. In *Proc. Sympos. Pure Math., Vol. V*, pages 1–27. American Mathematical Society, Providence, R.I., 1962.
16. Anne S. Troelstrs, editor. *Metamathematical Investigations of Intuitionistic Arithemtic and analysis*. Lecture Notes in Mathematics 344. Springer-Verlag, 1973.
17. Wim Veldman. Brouwer's fan theorem as an axiom and as a contrast to Kleene's alternative. Technical report, Department of Mathematics, Radboud University Nijmegen, 2005. Report No. 0509.
18. Wim Veldman. The principle of open induction on the unit interval [0,1] and some of its equivalents. Slides from presentation, May 2010.